

# Reduction of Error-Trellises for Tail-Biting Convolutional Codes Using Shifted Error-Subsequences

Masato Tajima

Graduate School of Sci. and Eng.  
University of Toyama

3190 Gofuku, Toyama 930-8555, Japan

Email: tajima@eng.u-toyama.ac.jp

Koji Okino

Information Technology Center  
University of Toyama

3190 Gofuku, Toyama 930-8555, Japan

Email: okino@itc.u-toyama.ac.jp

Tatsuto Murayama

Graduate School of Sci. and Eng.  
University of Toyama

3190 Gofuku, Toyama 930-8555, Japan

Email: murayama@eng.u-toyama.ac.jp

**Abstract**—In this paper, we discuss the reduction of error-trellises for tail-biting convolutional codes. In the case where some column of a parity-check matrix has a monomial factor  $D^l$ , we show that the associated tail-biting error-trellis can be reduced by cyclically shifting the corresponding error-subsequence by  $l$  (i.e., the power of  $D$ ) time units. We see that the resulting reduced error-trellis is again tail-biting. Moreover, we show that reduction is also possible using backward-shifted error-subsequences.

## I. INTRODUCTION

Tail-biting is a technique by which a convolutional code can be used to construct a block code without any loss of rate [4], [6], [14]. Let  $C_{tb}$  be a tail-biting convolutional code with an  $N$ -section code-trellis  $T_{tb}^{(c)}$ . The fundamental idea behind tail-biting is that the encoder starts and ends in the same state, i.e.,  $\beta_0 = \beta_N$  ( $\beta_k$  is the encoder state at time  $k$ ). Suppose that  $T_{tb}^{(c)}$  has  $\Sigma_0$  initial (or final) states, then it is composed of  $\Sigma_0$  subtrellises, each having the same initial and final states. We call these subtrellises tail-biting code subtrellises. For example, a tail-biting code-trellis of length  $N = 4$  based on the generator matrix

$$G_1(D) = (D + D^2, D^2, 1 + D) \quad (1)$$

is shown in Fig.1. Since  $\Sigma_0 = 4$ , this tail-biting code-trellis is composed of 4 code subtrellises. In Fig.1, bold lines correspond to the code subtrellis with  $\beta_0 = \beta_4 = (1, 1)$ . On the other hand, it is reasonable to think that an error-trellis  $T_{tb}^{(e)}$  for the tail-biting convolutional code  $C_{tb}$  can equally be constructed. In this case, each error subtrellis should have the same initial and final states like a code subtrellis. In our previous works [11], [12], taking this property into consideration, we have presented an error-trellis construction for tail-biting convolutional codes. For example, consider the above case. The parity-check matrix  $H_1(D)$  associated with  $G_1(D)$  is given by

$$H_1(D) = \begin{pmatrix} 1 & 0 & D \\ D & 1 + D & 0 \end{pmatrix}. \quad (2)$$

Let

$$z = z_1 z_2 z_3 z_4 = 110 101 101 011$$

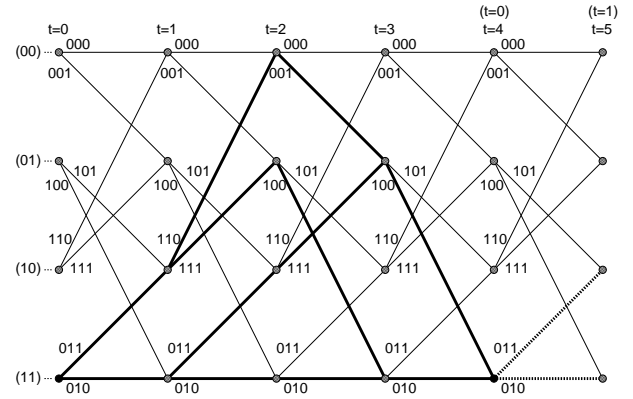


Fig. 1. Tail-biting code-trellis based on  $G_1(D)$ .

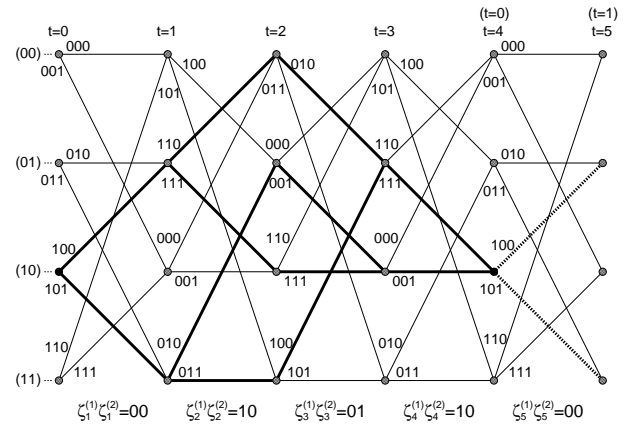


Fig. 2. Tail-biting error-trellis based on  $H_1^T(D)$ .

be the received data. In this case, using the method in [11], the tail-biting error-trellis corresponding to the code-trellis in Fig.1 can be constructed as is shown in Fig.2, where bold lines correspond to the error subtrellis with  $\sigma_0 = \sigma_4 = (1, 0)$ .

On the other hand [9], note that the third column of  $H_1(D)$  has the monomial factor  $D$ . Let  $e_k = (e_k^{(1)}, e_k^{(2)}, e_k^{(3)})$

and  $\zeta_k = (\zeta_k^{(1)}, \zeta_k^{(2)})$  be the time- $k$  error and syndrome, respectively. We have the following modification ( $T$  means transpose):

$$\zeta_k = e_k H_1^T(D) \quad (3)$$

$$\begin{aligned} &= (e_k^{(1)}, e_k^{(2)}, e_k^{(3)}) \begin{pmatrix} 1 & D \\ 0 & 1+D \\ D & 0 \end{pmatrix} \\ &= (e_k^{(1)}, e_k^{(2)}, De_k^{(3)}) \begin{pmatrix} 1 & D \\ 0 & 1+D \\ 1 & 0 \end{pmatrix} \\ &\triangleq \tilde{e}_k \tilde{H}_1^T(D), \end{aligned} \quad (4)$$

where  $\tilde{e}_k = (e_k^{(1)}, e_k^{(2)}, \tilde{e}_k^{(3)})$  and  $\tilde{e}_k^{(3)}$  is defined as  $De_k^{(3)} = e_{k-1}^{(3)}$ . Since the overall constraint length  $\tilde{\nu}^\perp$  of

$$\tilde{H}_1(D) = \begin{pmatrix} 1 & 0 & 1 \\ D & 1+D & 0 \end{pmatrix} \quad (5)$$

is one, the above equation implies that the tail-biting error-trellis in Fig.2 can be reduced by shifting the subsequence  $\{e_k^{(3)}\}$  by the unit time.

In this paper, taking the above example into account, we discuss the reduction of error-trellises for tail-biting convolutional codes. It is assumed that some ( $j$ th) column of a parity-check matrix  $H(D)$  has a monomial factor  $D^{l_j}$ . In this case, we show that the associated tail-biting error-trellis can be reduced by cyclically shifting the  $j$ th component  $e_k^{(j)}$  by  $l_j$  time units. We also show that the resulting reduced error-trellis is again tail-biting. We see that a kind of “periodicity” inherent in tail-biting trellises plays a key role in our discussion.

## II. PRELIMINARIES

In this paper, we always assume that the underlying field is  $F = \text{GF}(2)$ . Let  $G(D)$  be a generator matrix of an  $(n, n-m)$  convolutional code  $C$ . Let  $H(D)$  be a corresponding  $m \times n$  parity-check matrix of  $C$ . Both  $G(D)$  and  $H(D)$  are assumed to be canonical [1], [5]. Denote by  $\nu^\perp$  the overall constraint length of  $H(D)$  and by  $M$  the memory length of  $H(D)$  (i.e., the maximum degree among the polynomials of  $H(D)$ ). Then  $H(D)$  is expressed as

$$H(D) = H_0 + H_1 D + \cdots + H_M D^M. \quad (6)$$

### A. Adjoint-Obvious Realization of a Syndrome Former

Consider the adjoint-obvious realization (observer canonical form [2], [3]) of the syndrome former  $H^T(D)$ . Let  $e_k = (e_k^{(1)}, e_k^{(2)}, \dots, e_k^{(n)})$  and  $\zeta_k = (\zeta_k^{(1)}, \zeta_k^{(2)}, \dots, \zeta_k^{(m)})$  be the input error and the corresponding output syndrome at time  $k$ , respectively. Denote by  $\sigma_{kp}^{(q)}$  the contents of the memory elements in the above realization. (If a memory element is missing, the corresponding  $\sigma_{kp}^{(q)}$  is set to zero.) Using  $\sigma_{kp}^{(q)}$ , the syndrome-former state at time  $k$  is defined as

$$\sigma_k \triangleq (\sigma_{k1}^{(1)}, \dots, \sigma_{k1}^{(m)}, \dots, \sigma_{kM}^{(1)}, \dots, \sigma_{kM}^{(m)}). \quad (7)$$

(Remark: The effective size of  $\sigma_k$  is equal to  $\nu^\perp$ .)

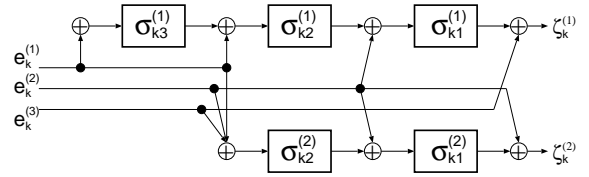


Fig. 3. Adjoint-obvious realization (observer canonical form) of syndrome former  $H_2^T(D)$ .

For example, Fig.3 illustrates the adjoint-obvious realization of the syndrome former  $H_2^T(D)$  [1], where

$$H_2(D) = \begin{pmatrix} D^2 + D^3 & D & 1 \\ D^2 & 1 + D + D^2 & D^2 \end{pmatrix}. \quad (8)$$

Hence, we have

$$\sigma_k = (\sigma_{k1}^{(1)}, \sigma_{k1}^{(2)}, \sigma_{k2}^{(1)}, \sigma_{k2}^{(2)}, \sigma_{k3}^{(1)}, 0). \quad (9)$$

Note that the effective size of  $\sigma_k$  is  $\nu^\perp = 5$ .

Under the above conditions [7], [8], we have

$$\begin{aligned} \sigma_k &= (\sigma_k^{(1)}, \sigma_k^{(2)}, \dots, \sigma_k^{(M)}) \\ &= (e_{k-M+1}, \dots, e_{k-1}, e_k) \\ &\quad \times \begin{pmatrix} H_M^T & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots \\ H_2^T & \dots & H_M^T & 0 \\ H_1^T & \dots & H_{M-1}^T & H_M^T \end{pmatrix}. \end{aligned} \quad (10)$$

Note that  $\sigma_k$  has an alternative expression:

$$\sigma_k = (\sigma_{k-1}^{(2)}, \dots, \sigma_{k-1}^{(M)}, 0) + e_k (H_1^T, H_2^T, \dots, H_M^T). \quad (11)$$

Similarly,  $\zeta_k$  is expressed as

$$\zeta_k = e_{k-M} H_M^T + \cdots + e_{k-1} H_1^T + e_k H_0^T \quad (12)$$

$$= \sigma_{k-1}^{(1)} + e_k H_0^T. \quad (13)$$

### B. Dual States

The encoder states can be labeled by the syndrome-former states (i.e., dual states [2]). The dual state  $\beta_k^*$  corresponding to an encoder state  $\beta_k$  is obtained by replacing  $e_k$  in  $\sigma_k$  by  $y_k = u_k G(D)$  ( $y_k$  is the code symbol at time  $k$  corresponding to the information symbol  $u_k$ ). We have

$$\begin{aligned} \beta_k^* &= (y_{k-M+1}, \dots, y_{k-1}, y_k) \\ &\quad \times \begin{pmatrix} H_M^T & \dots & 0 & 0 \\ \dots & \dots & \dots & \dots \\ H_2^T & \dots & H_M^T & 0 \\ H_1^T & \dots & H_{M-1}^T & H_M^T \end{pmatrix}. \end{aligned} \quad (14)$$

*Example 1:* Consider the parity-check matrix  $H_1(D)$ . We have

$$\begin{aligned} H_1(D) &= \begin{pmatrix} 1 & 0 & D \\ D & 1+D & 0 \end{pmatrix} \\ &= \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 \\ 1 & 1 & 0 \end{pmatrix} D \\ &\triangleq H_0 + H_1 D. \end{aligned} \quad (15)$$

Then the dual state corresponding to an encoder state  $\beta_k = (u_{k-1}, u_k)$  is obtained as follows.

$$\begin{aligned}\beta_k^* &= \mathbf{y}_k H_1^T \\ &= (y_k^{(1)}, y_k^{(2)}, y_k^{(3)}) \begin{pmatrix} 0 & 1 \\ 0 & 1 \\ 1 & 0 \end{pmatrix} \\ &= (y_k^{(3)}, y_k^{(1)} + y_k^{(2)}) \\ &= (u_{k-1} + u_k, u_{k-1}).\end{aligned}\quad (16)$$

### C. Error-Trellises for Tail-Biting Convolutional Codes

Suppose that a tail-biting code-trellis based on  $G(D)$  is defined in  $[0, N]$ , where  $N \geq M$ . In this case, the corresponding tail-biting error-trellis based on  $H^T(D)$  is constructed as follows [11].

*Step 1:* Let  $\mathbf{z} = \{z_k\}_{k=1}^N$  be a received data. Denote by  $\sigma_0$  the initial state of the syndrome former  $H^T(D)$ . Let  $\sigma_{fin} (= \sigma_N)$  be the final syndrome-former state corresponding to the input  $\mathbf{z}$ . Note that  $\sigma_{fin}$  is independent of  $\sigma_0$  and is uniquely determined only by  $\mathbf{z}$ .

*Step 2:* Set  $\sigma_0$  (i.e., the initial state of the syndrome former) to  $\sigma_{fin}$  and input  $\mathbf{z}$  to the syndrome former again. Here, suppose that the syndrome sequence  $\zeta = \{\zeta_k\}_{k=1}^N$  is obtained. (Remark:  $\zeta_k$  ( $k \geq M+1$ ) has been obtained in Step 1.)

*Step 3:* Concatenate the error-trellis modules corresponding to the syndromes  $\zeta_k$ . Then we have the tail-biting error-trellis.

*Example 2:* Again, consider the parity-check matrix  $H_1(D)$ . Let

$$\mathbf{z} = z_1 z_2 z_3 z_4 = 110 101 101 011 \quad (17)$$

be the received data. According to Step 1, let us input  $\mathbf{z}$  to the syndrome former  $H_1^T(D)$ . Then we have  $\sigma_{fin} = (1, 1)$ . Next, set  $\sigma_0$  to  $\sigma_{fin} = (1, 1)$  and input  $\mathbf{z}$  to the syndrome former again. In this case, the syndrome sequence

$$\zeta = \zeta_1 \zeta_2 \zeta_3 \zeta_4 = 00 10 01 10 \quad (18)$$

is obtained. The tail-biting error-trellis is constructed by concatenating the error-trellis modules corresponding to  $\zeta_k$ . The resulting tail-biting error-trellis is shown in Fig.2.

With respect to the correspondence between tail-biting code subtrellises and tail-biting error subtrellises, we have the following [11], [12].

*Proposition 1:* Let  $\beta_0 (= \beta_N) = \beta$  be the initial (final) state of a tail-biting code subtrellis. Then the initial (final) state of the corresponding tail-biting error subtrellis is given by  $\sigma_{fin} + \beta^*$ .

*Example 2 (Continued):* Consider the tail-biting error-trellis in Fig.2. In this example, we have  $\sigma_{fin} = (1, 1)$ . The corresponding tail-biting code-trellis based on  $G_1(D)$  is shown in Fig.1. In Fig.1, take notice of the code subtrellis with initial (final) state  $\beta = (1, 1)$  (bold lines). The dual state of  $\beta = (1, 1)$  is calculated as  $\beta^* = (u_{-1} + u_0, u_{-1}) = (1 + 1, 1) = (0, 1)$ . Hence, the initial (final) state of the corresponding error subtrellis is given by  $\sigma_{fin} + \beta^* = (1, 1) + (0, 1) = (1, 0)$  (bold lines in Fig.2).

## III. REDUCTION OF TAIL-BITING ERROR-TRELLISES

### A. Error-Trellis Reduction Using Shifted Error-Subsequences

Consider the example in Section I. Noting the relation  $\tilde{e}_k^{(3)} = e_{k-1}^{(3)}$ , we cyclically shift the third component of each  $\mathbf{z}_k$  to the right by the unit time. Then we have the modified received data

$$\tilde{\mathbf{z}} = \tilde{z}_1 \tilde{z}_2 \tilde{z}_3 \tilde{z}_4 = 111 100 101 011. \quad (19)$$

Applying the method in Section II-C, we can construct a reduced tail-biting error-trellis. According to Step 1, let us input  $\tilde{\mathbf{z}}$  to the syndrome former  $\tilde{H}_1^T(D)$ . Then we have  $\tilde{\sigma}_{fin} = (1)$ . Next, set  $\tilde{\sigma}_0$  to  $\tilde{\sigma}_{fin} = (1)$  and input  $\tilde{\mathbf{z}}$  to the syndrome former again. In this case, the same syndrome sequence as the original one (i.e.,  $\zeta = 00 10 01 10$ ) is obtained. The reduced tail-biting error-trellis is constructed by concatenating the reduced error-trellis modules corresponding to  $\zeta_k$ . The resulting tail-biting error-trellis is shown in Fig.4. Here let us examine how a tail-biting error subtrellis is embedded in the corresponding reduced error-trellis. For the purpose, take notice of the subtrellis with initial (final) state  $(1, 0)$  (bold lines in Fig.2). First, consider where the state  $(1, 0)$  is mapped to. In the original error-trellis, the final state  $\sigma_N$  is expressed as

$$\sigma_N = e_N H_1^T = (e_N^{(3)}, e_N^{(1)} + e_N^{(2)}). \quad (20)$$

Using the relation  $e_N^{(3)} = \tilde{e}_{N+1}^{(3)}$ ,  $\sigma_N = (e_N^{(3)}, e_N^{(1)} + e_N^{(2)})$  is modified as  $(\tilde{e}_{N+1}^{(3)}, e_N^{(1)} + e_N^{(2)})$ . Since the subscript  $N+1$  ( $> N$ ) is inappropriate for the state at time  $N$ , we have

$$\tilde{\sigma}_N = e_N^{(1)} + e_N^{(2)} = (\tilde{e}_N \tilde{H}_1^T). \quad (21)$$

(Remark: We have  $\tilde{e}_N \tilde{H}_1^T = (0, e_N^{(1)} + e_N^{(2)})$ . Hence, the first component can be deleted.) That is, state  $\sigma_4 = (1, 0)$  is mapped to  $\tilde{\sigma}_4 = (0)$ .

Next, consider an arbitrary error-path

$$\mathbf{e}_p = e_1 e_2 e_3 e_4$$

in the subtrellis with initial (final) state  $(1, 0)$ . Here take notice of two sections from  $t = 0$  to  $t = 1$  and from  $t = 3$  to  $t = 4$ . Note that these are adjacent sections in the circular error-trellis. From  $\sigma_4 = (e_4^{(3)}, e_4^{(1)} + e_4^{(2)}) = (1, 0)$ , we have  $e_4^{(3)} = 1$ . Since the third component of each  $\mathbf{e}_k$  is cyclically shifted to the right by the unit time,  $e_1^{(3)}$  is replaced by  $e_4^{(3)} = 1$ . That is, the third label on the first branch of the error-path in the reduced trellis must be 1. By taking account of these conditions, we have four admissible error-paths:

$$\begin{aligned}\tilde{\mathbf{e}}_{p_1} &= 101 110 010 110 \\ \tilde{\mathbf{e}}_{p_2} &= 101 110 111 001 \\ \tilde{\mathbf{e}}_{p_3} &= 101 011 000 001 \\ \tilde{\mathbf{e}}_{p_4} &= 101 011 101 110.\end{aligned}$$

These paths are denoted with bold lines in Fig.4. Since a planar trellis is used in Fig.4, the first segment (i.e., 101) of each error-path is added to it as a tail. If a circular trellis is used, this augmentation is unnecessary. In this way, we obtain the

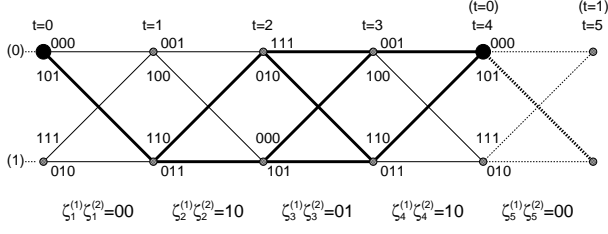


Fig. 4. Reduced tail-biting error-trellis based on  $\tilde{H}_1^T(D)$ .

reduced tail-biting error subtrellis. The original error-paths are restored by noting the relation  $e_k^{(3)} = \tilde{e}_{k+1}^{(3)}$ . That is, we only need to cyclically shift the third component of each  $\tilde{e}_k = (e_k^{(1)}, e_k^{(2)}, \tilde{e}_k^{(3)})$  to the left by the unit time. As a result, four error-paths

$$\begin{aligned} e_{q_1} &= 100\ 110\ 010\ 111 \\ e_{q_2} &= 100\ 111\ 111\ 001 \\ e_{q_3} &= 101\ 010\ 001\ 001 \\ e_{q_4} &= 101\ 011\ 100\ 111 \end{aligned}$$

are obtained. We see that these paths completely coincide with those in Fig.2.

### B. General Cases

The argument in the previous subsection, though it was presented in terms of a specific example, is entirely general. Hence, it can be directly extended to general cases. Suppose that a specific ( $j$ th) column of  $H(D)$  has the form

$$\begin{pmatrix} D^{l_j} \tilde{h}_{1j}(D) & D^{l_j} \tilde{h}_{2j}(D) & \dots & D^{l_j} \tilde{h}_{mj}(D) \end{pmatrix}^T, \quad (22)$$

where  $1 \leq l_j \leq M$ . (Remark: A more general case where each column has the above form can also be treated.) Let  $\tilde{H}(D)$  be the modified version of  $H(D)$  with the  $j$ th column being replaced by

$$\begin{pmatrix} \tilde{h}_{1j}(D) & \tilde{h}_{2j}(D) & \dots & \tilde{h}_{mj}(D) \end{pmatrix}^T. \quad (23)$$

$\tilde{H}(D)$  is assumed to be canonical. In this case, the reduction of a tail-biting error-trellis is accomplished as follows.

(i) *Fundamental relation:* Denote by  $e_k = (e_k^{(1)}, \dots, e_k^{(n)})$  and  $\zeta_k = (\zeta_k^{(1)}, \dots, \zeta_k^{(m)})$  the time- $k$  error and syndrome, respectively. Also, let  $\tilde{e}_k \triangleq (e_k^{(1)}, \dots, e_{<k-l_j>}^{(j)}, \dots, e_k^{(n)})$ , where  $<t>$  denotes  $t \bmod N$ . Then we have

$$\zeta_k = \tilde{e}_k \tilde{H}^T(D). \quad (24)$$

(ii) *Construction of reduced tail-biting error-trellises:* Let

$$z = \{z_k\}_{k=1}^N = \{(z_k^{(1)}, \dots, z_k^{(j)}, \dots, z_k^{(n)})\}_{k=1}^N \quad (25)$$

be a received data. We construct the modified received data

$$\tilde{z} = \{\tilde{z}_k\}_{k=1}^N \triangleq \{(z_k^{(1)}, \dots, z_{<k-l_j>}^{(j)}, \dots, z_k^{(n)})\}_{k=1}^N \quad (26)$$

by cyclically shifting the  $j$ th component of each  $z_k$  to the right by  $l_j$  time units. By applying the method in Section II-C to the modified syndrome former  $\tilde{H}^T(D)$  and the modified received

data  $\tilde{z}$ , a reduced tail-biting error-trellis is constructed. Note that the same syndrome sequence  $\{\zeta_k\}$  as for the tail-biting error-trellis based on  $H^T(D)$  is obtained.

(iii) *Reduced tail-biting error subtrellises:* Let  $ST_{tb}^{(e)}$  be a tail-biting error subtrellis with initial (final) state  $\sigma_N$ .  $\sigma_N$  can be expressed using  $\{e_t\}_{t=N-M+1}^N$  (cf. (10)). Here replace each  $e_t^{(j)}$  ( $N-M+1 \leq t \leq N$ ) by  $\tilde{e}_{t+l_j}^{(j)}$  and delete those terms  $\tilde{e}_t^{(j)}$  with subscript  $t$  greater than  $N$ . Denote by  $\tilde{\sigma}_N$  the resulting state expression. In this case, state  $\sigma_N$  is mapped to state  $\tilde{\sigma}_N$  in the reduced tail-biting error-trellis.

Consider the two trellis-sections from  $t=0$  to  $t=l_j$  and from  $t=N-l_j$  to  $t=N$ . Note that these form a continuous section of length  $2l_j$  in the circular error-trellis. Now we can solve Eq.(10) ( $k=N$ ) given  $\sigma_N$ . (Remark:  $\{e_t^{(j)}\}_{t=N-l_j+1}^N$  is uniquely determined under a moderate condition on  $H(D)$ .) Since the  $j$ th component of each  $e_k$  is cyclically shifted to the right by  $l_j$  time units,  $e_t^{(j)}$  ( $1 \leq t \leq l_j$ ) is replaced by  $e_{N-l_j+t}^{(j)}$ . That is, the  $j$ th component  $\tilde{e}_t^{(j)}$  of the reduced path-segment  $\tilde{e}_t$  ( $1 \leq t \leq l_j$ ) must be  $e_{N-l_j+t}^{(j)}$ . We call these segments “admissible”. Then  $ST_{tb}^{(e)}$  is embedded in the reduced tail-biting error subtrellis with initial (final) state  $\tilde{\sigma}_N$ , where the path-segments in the first  $l_j$  sections are restricted to admissible ones.

(iv) *Restoration of the original error-paths:* The original error-paths are restored by noting the relation  $e_k^{(j)} = \tilde{e}_{<k+l_j>}^{(j)}$ . That is, for an error-path

$$\tilde{e} = \{\tilde{e}_k\}_{k=1}^N = \{(e_k^{(1)}, \dots, \tilde{e}_k^{(j)}, \dots, e_k^{(n)})\}_{k=1}^N, \quad (27)$$

we only need to cyclically shift the  $j$ th component of each  $\tilde{e}_k$  to the left by  $l_j$  time units.

We remark that  $z$  has been periodically extended in both directions and this periodicity is fully used for tail-biting error-trellis construction. Now the relation  $\zeta_k = e_k H^T(D)$  is equivalently modified as  $\zeta_k = \tilde{e}_k \tilde{H}^T(D)$ . Note that the correspondence between  $\{e_k\}$  and  $\{\tilde{e}_k\}$  is one-to-one ( $\{e_k^{(j)}\}$  is cyclically shifted). Hence, the original error-path  $e = \{e_k\}$  is indirectly represented using the reduced tail-biting error-trellis based on  $\tilde{H}^T(D)$ . (Accordingly, the restoration in (iv) is required.) Notice that the overall constraint length  $\tilde{\nu}^\perp$  of  $\tilde{H}(D)$  is not more than  $\nu^\perp$ . Thus we have shown the following.

*Proposition 2:* Let  $T_{tb}^{(e)}$  be a tail-biting error-trellis based on  $H^T(D)$ , where the  $j$ th column of  $H(D)$  has a monomial factor  $D^{l_j}$ . Also, suppose that  $\tilde{\nu}^\perp < \nu^\perp$ . Then  $T_{tb}^{(e)}$  can be reduced by cyclically shifting the  $j$ th subsequence of  $\{e_k\}$  by  $l_j$  time units. In this case, the reduced error-trellis  $\tilde{T}_{tb}^{(e)}$  is again tail-biting.

### C. Error-Trellis Reduction Using Backward-Shifted Error-Subsequences

A reduced tail-biting error-trellis can be constructed not only using forward-shifted error-subsequences but also using “backward-shifted” error-subsequences [9]. For example, con-

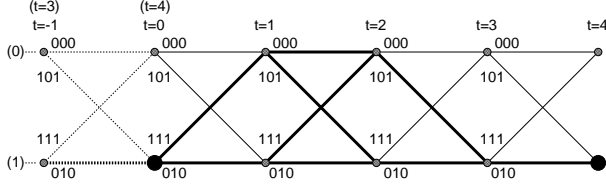


Fig. 5. Reduced tail-biting code-trellis based on  $\tilde{G}_1(D)$ .

sider the parity-check matrix in (8):

$$H_2(D) = \begin{pmatrix} D^2 + D^3 & D & 1 \\ D^2 & 1 + D + D^2 & D^2 \end{pmatrix}.$$

Since, the first column has the monomial factor  $D^2$ ,  $H_2(D)$  can be reduced to

$$\tilde{H}_2(D) = \begin{pmatrix} 1 + D & D & 1 \\ 1 & 1 + D + D^2 & D^2 \end{pmatrix} \quad (28)$$

by dividing the first column of  $H_2(D)$  by  $D^2$ . On the other hand, we can “multiply” the second and third columns by  $D^2$ . Note that this corresponds to backward-shifting by two time units in terms of the original  $\{e_k^{(j)}\}$  ( $j = 2, 3$ ) and we have

$$H'_2(D) = \begin{pmatrix} D^2 + D^3 & D^3 & D^2 \\ D^2 & D^2 + D^3 + D^4 & D^4 \end{pmatrix}. \quad (29)$$

This matrix can be reduced to an equivalent canonical parity-check matrix  $\tilde{H}_2(D)$ . (Note that the first and second rows of  $H'_2(D)$  are just delayed versions of the first and second rows of  $\tilde{H}_2(D)$ .)

#### IV. REDUCTION OF TAIL-BITING CODE-TRELLISES

A code-trellis for a tail-biting convolutional code and the corresponding error-trellis are dual to each other. Hence, the reduction of tail-biting code-trellises is also possible. For example, consider the generator matrix  $G_1(D)$  in (1). Observe that the first and second columns of  $G_1(D)$  have the monomial factor  $D$ . This fact enables reduction of the original tail-biting code-trellis. Let  $u_k$  and  $\mathbf{y}_k = (y_k^{(1)}, y_k^{(2)}, y_k^{(3)})$  be the information and code symbol at time  $k$ , respectively. Then the relation  $\mathbf{y}_k = u_k G_1(D)$  is equivalently modified as  $\tilde{\mathbf{y}}_k = u_k \tilde{G}_1(D)$ . Here,

$$\tilde{\mathbf{y}}_k = (\tilde{y}_k^{(1)}, \tilde{y}_k^{(2)}, \tilde{y}_k^{(3)}) \triangleq (y_{k+1}^{(1)}, y_{k+1}^{(2)}, y_k^{(3)}) \quad (30)$$

and  $\tilde{G}_1(D)$  is defined as

$$\tilde{G}_1(D) = (1 + D, D, 1 + D). \quad (31)$$

Using a similar argument as that in Section III-A, a reduced tail-biting code-trellis associated with the one in Fig.1 is constructed. The resulting reduced code-trellis is shown in Fig.5, where bold lines correspond to the original code subtrellis with  $\beta_0 = \beta_4 = (1, 1)$ . Note that the first two labels on each branch of the error-path are shifted to the left (i.e., backward-shifted) by the unit time. Accordingly, the path-segment from  $t = 3$  to  $t = 4$  is restricted to 010. We see that this specific example can be directly extended to general cases. We also

remark that the reduction of a tail-biting code-trellis and that of the corresponding tail-biting error-trellis can be accomplished simultaneously, if reduction is possible (cf. [10]).

#### V. CONCLUSION

In this paper, we have discussed the reduction of error-trellises for tail-biting convolutional codes. In the case where a given parity-check matrix  $H(D)$  has a monomial factor  $D^l$  in some column, we have shown that the associated tail-biting error-trellis can be reduced by cyclically shifting the corresponding error-subsequence by  $l$  time units. We have also shown that the obtained reduced error-trellis is again tail-biting. Moreover, we have shown that trellis-reduction is also accomplished using backward-shifted error-subsequences. The proposed method has been applied to concrete examples and it has been confirmed that each subtrellis is successfully embedded in the reduced tail-biting error-trellis. Finally, we have shown that the associated tail-biting code-trellis can equally be reduced using shifted code-subsequences. We remark that the convolutional code specified by a parity-check matrix  $H(D)$  with the form discussed in the paper has a relatively poor distance property. We also remark that such parity-check matrices appear, for example, in [13].

#### REFERENCES

- [1] M. Ariel and J. Snyders, “Error-trellises for convolutional codes—Part I: Construction,” *IEEE Trans. Commun.*, vol. 46, no. 12, pp. 1592–1601, Dec. 1998.
- [2] G. D. Forney, Jr., “Structural analysis of convolutional codes via dual codes,” *IEEE Trans. Inform. Theory*, vol. IT-19, no. 4, pp. 512–518, July 1973.
- [3] R. Johannesson and K. S. Zigangirov, *Fundamentals of Convolutional Coding*. New York: IEEE Press, 1999.
- [4] H. H. Ma and J. K. Wolf, “On tail biting convolutional codes,” *IEEE Trans. Commun.*, vol. COM-34, no. 2, pp. 104–111, Feb. 1986.
- [5] R. J. McEliece and W. Lin, “The trellis complexity of convolutional codes,” *IEEE Trans. Inform. Theory*, vol. 42, no. 6, pp. 1855–1864, Nov. 1996.
- [6] R. Y. Shao, S. Lin, and M. P. C. Fossorier, “Two decoding algorithms for tailbiting codes,” *IEEE Trans. Commun.*, vol. 51, no. 10, pp. 1658–1665, Oct. 2003.
- [7] V. Sidorenko and V. Zyablov, “Decoding of convolutional codes using a syndrome trellis,” *IEEE Trans. Inform. Theory*, vol. 40, no. 5, pp. 1663–1666, Sept. 1994.
- [8] M. Tajima, K. Okino, and T. Miyagoshi, “Minimal code(error)-trellis module construction for rate- $k/n$  convolutional codes: Extension of Yamada-Harashima-Miyakawa’s construction,” *IEICE Trans. Fundamentals*, vol. E90-A, no. 11, pp. 2629–2634, Nov. 2007.
- [9] ———, “Error-trellis construction for convolutional codes using shifted error/syndrome-subsequences,” *IEICE Trans. Fundamentals*, vol. E92-A, no. 8, pp. 2086–2096, Aug. 2009.
- [10] ———, “Simultaneous code/error-trellis reduction for convolutional codes using shifted code/error-subsequences,” in *Proc. 2011 IEEE Int. Symp. Inf. Theory*, pp. 2409–2413, Aug. 2011.
- [11] ———, “Error-trellis construction for tailbiting convolutional codes,” *IEICE Technical Report*, IT2011-59, pp. 85–90, March 2012.
- [12] M. Tajima and K. Okino, “Error-trellises for tailbiting convolutional codes,” in *Proc. ISITA2012*, pp. 648–652, Oct. 2012.
- [13] R. M. Tanner, D. Sridhara, A. Sridharan, T. E. Fuja, and D. J. Costello, Jr., “LDPC block and convolutional codes based on circulant matrices,” *IEEE Trans. Inform. Theory*, vol. 50, no. 12, pp. 2966–2984, Dec. 2004.
- [14] M. B. S. Tavares, K. S. Zigangirov, and G. P. Fettweis, “Tail-biting LDPC convolutional codes,” in *Proc. 2007 IEEE Int. Symp. Inf. Theory*, pp. 2341–2345, June 2007.